# Welcome to Assignment 5

There are two parts in this assignment. The conceptual questions could be answered by words, and the coding questions require some computation.

## Conceptual questions

### Q1. Suppose we want to build a connected network with 10 nodes and average degree 1.8, such that the heterogeneity parameter is largest. What does the graph look like?

hint: you don't have to use Networkx to construct a network and plot it. A simple description would do :)

We have ten nodes, and the average degree is 1.8. To maximize the heterogeneity parameter, we need a graph with hubs. The presence of a hub ensures that the heterogeneity parameter will be high, as they are directly correlated. Also, since the average degree is close to 2 but slightly lesser, we need nodes with degrees 2 or less than 2, except for the hub. The hub will inflate the average degree value with its high degree.

Therefore, the network will have a high-degree hub and nodes with very low degrees.

### Q2. Are there networks such that the average number of neighbors of a node's neighbors match the average degree? If there are, what property must they have?

Yes, there are networks where the average number of neighbors of a node's neighbors matches the average degree. For this to hold true, we need a network where the high-degree nodes are connected to other high-degree nodes, and the low-degree nodes are connected to other low-degree nodes. This type of behavior can be observed in an assortative network. Thus, the graph needs to be an assortative network.

## Coding Questions

We will be using **socfb-Northwestern25** network for the following questions. You could find it in the book's Github repository dataset folder or on Canvas. It is a snapshot of Northwestern University's Facebook network. The nodes are anonymous users and the links are friend relationships. Load this network into a NetworkX graph in order to answer the following questions. **Be sure to use the proper graph class for an undirected, unweighted network.**

In [1]:
```
import numpy as np
import os
import networkx as nx
```

In [2]:
```
current_directory = os.getcwd()
# move back a directory
parent_dir = os.path.dirname(current_directory)

G = nx.read_edgelist(parent_dir + '/datasets/socfb-Northwestern25/socfb-Northwestern25.edges.gz')
nx.info(G)
```

```
/var/folders/_3/98b473cn1k993m8ln5gwyv6h0000gn/T/ipykernel_80631/2719763623.py:6: DeprecationWarning: info is deprecated and will be removed in version 3.0.

  nx.info(G)
```

Out[2]:
```
'Graph with 10567 nodes and 488337 edges'
```

### Q3. What proportion of nodes have degree 100 or greater?

hint: ratio of number of nodes having degree no less than 100 divided by the total number of nodes.

In [3]:
```
total_nodes = G.number_of_nodes()
total_nodes
```

Out[3]:
```
10567
```

In [4]:
```
degree_100greater = 0

all_degrees = dict(G.degree())

for node,degree in all_degrees.items() :
    if degree >= 100:
        degree_100greater += 1
```

In [5]:
```
proportion = degree_100greater/total_nodes
print(f'Proportion of nodes have degree 100 or greater: {round(proportion, 4)}')
```

```
Proportion of nodes have degree 100 or greater: 0.3821
```

### Q4. What is the maximum degree in this network?

### Q5. Users in this network are anonymized by giving the nodes numerical names. Which node has the highest degree?

In [6]:
```
highest_degree = max(G.degree(), key=lambda x: x[1])[1]
nodes_highest_degree = [node for node, degree in G.degree() if degree == highest_degree]

print(f'Highest degree: {highest_degree}')
print(f'Nodes with highest degree: {nodes_highest_degree}')
```

```
Highest degree: 2105
Nodes with highest degree: ['8262']
```

We check if there are multiple nodes/users with the highest degree. But as we observe, there is only one user: 8262 having the maximum degree.

### Q6. What is 95th percentile for degree, i.e., the value such that 95% of nodes have this degree or less?

In [7]:
```
degrees = list(all_degrees.values())
degrees_sorted = sorted(degrees)


# 95th percentile index
index_95 = int(np.ceil(0.95 * len(degrees_sorted)))
percentile_95 = degrees_sorted[index_95 - 1]

print(f'95th percentile for degree: {percentile_95}')
```

```
95th percentile for degree: 244
```

### Q7. Is the network assortative or disassortative?

In [8]:
```
assortativity_coef = nx.degree_assortativity_coefficient(G)
print(round(assortativity_coef, 4))
```

```
0.0344
```

Here the assortativity coefficient is positive and thus, we can say that the network is assortative. This means high-degree nodes are connected to other high-degree nodes, and low-degree nodes are connected to other low-degree nodes

### Q8. What is the mean degree for nodes in this network?

In [9]:
```
mean_degree = sum(degrees)/len(degrees)

print(f"Mean degree: {round(mean_degree, 4)}")
```

```
Mean degree: 92.4268
```

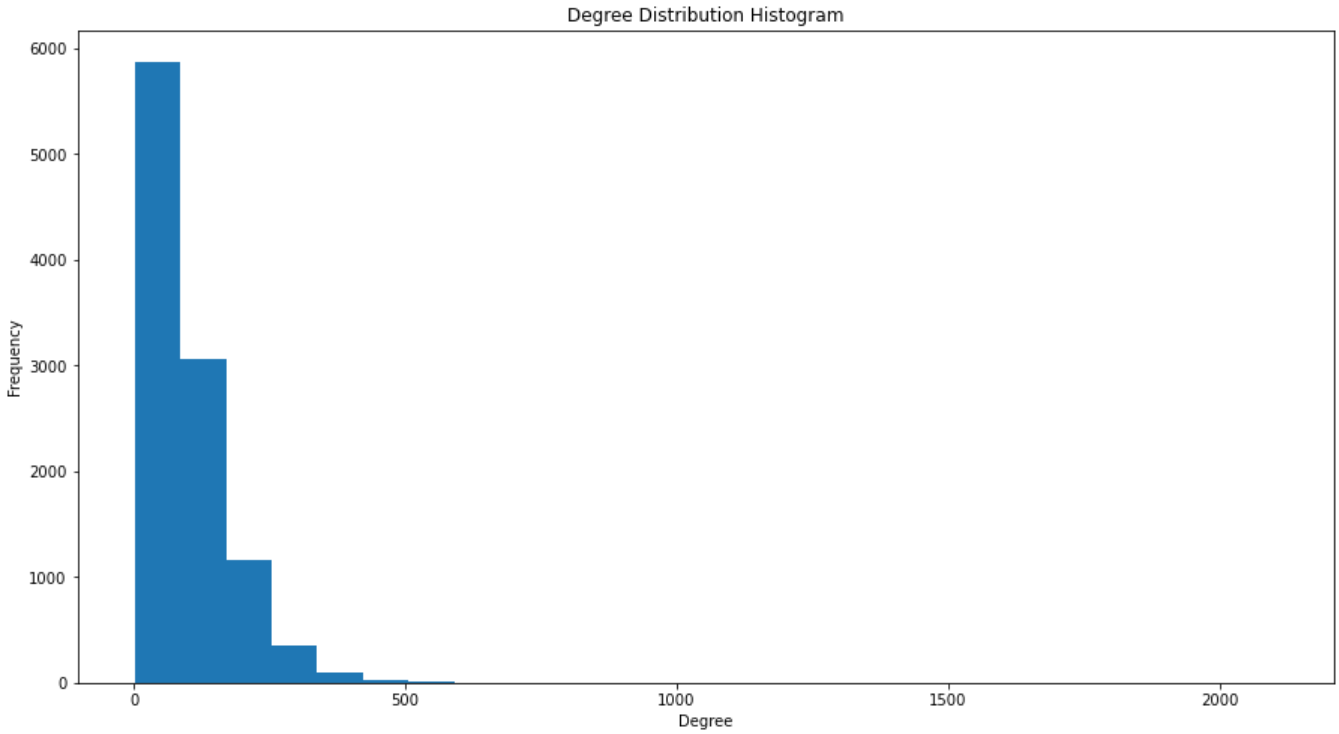
### Q9. Which of the following shapes best describes the degree distribution in this network? You can obtain the answer visually using tools we have introduced such as histograms, cumulatives, or log-scale plots.

a) Normal
b) Poissonian
c) Power law

d) Exponential

Hint: 1) At least one visualization is needed 2) select your answer from the four options based on your visualizations

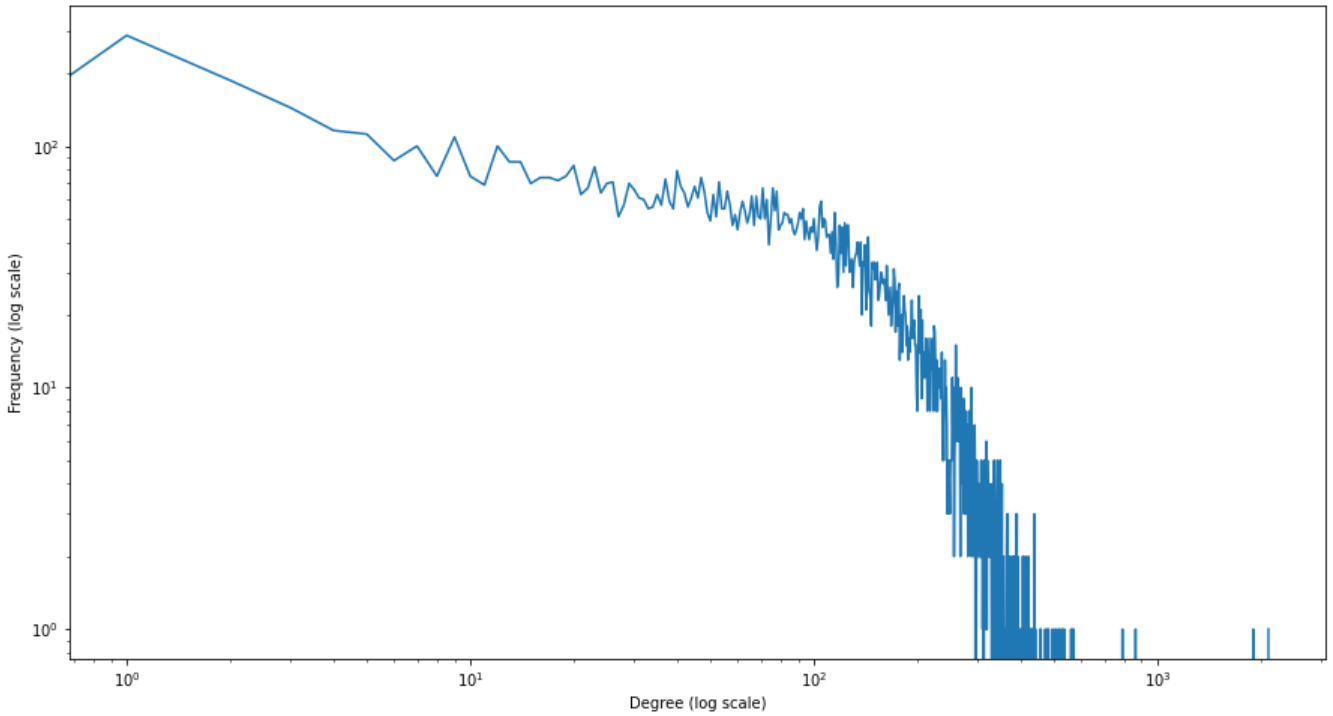
In [10]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(15, 8))
plt.hist(degrees, bins=25)
plt.title('Degree Distribution Histogram')
plt.xlabel('Degree')
plt.ylabel('Frequency')
plt.show()
```




The above plot has a peak and a skewed normal distribution. The right tail of the plot is very long and it seems like an Exponential distribution. Taking a look at the log scale plot of the same will shed more light.

In [11]:

```
# Plot log-log plot
degree_counts = nx.degree_histogram(G)
degrees_range = range(len(degree_counts))
plt.figure(figsize=(15, 8))
plt.loglog(degrees_range, degree_counts)
plt.xlabel('Degree (log scale)')
plt.ylabel('Frequency (log scale)')
plt.show()
```




In [ ]:

```

```